Supporting Advanced Video Formats

Session 208

These are confidential sessions—please refrain from filming or taking pictures.

Graphics and Media

Supporting Advanced Video Formats

Sam Bushell
QuickTime Engineering

http://developer.apple.com/wwdc2005/

What Is This Session About?

- State of the art codecs in QuickTime 7
  - H.264
- Technology background
- How to enable H.264 in your applications
  - Modern APIs
  - Sample code!

H.264

- Joint effort on video codec from organizations that brought us MPEG and H.261 / H.263
- Standardized in 2003
- Chosen for HD-DVD, Blu-Ray, and 3GPP standards
- Flexible
  - From 3GPP (176x144 @ 64 kbit/s)
  - To HD (1920x1080 @ 8 Mbit/s)
- Known as MPEG-4 Part 10, JVT, H.264, and AVC
- New video codec in QuickTime 7

Demo

H.264

http://developer.apple.com/wwdc2005/

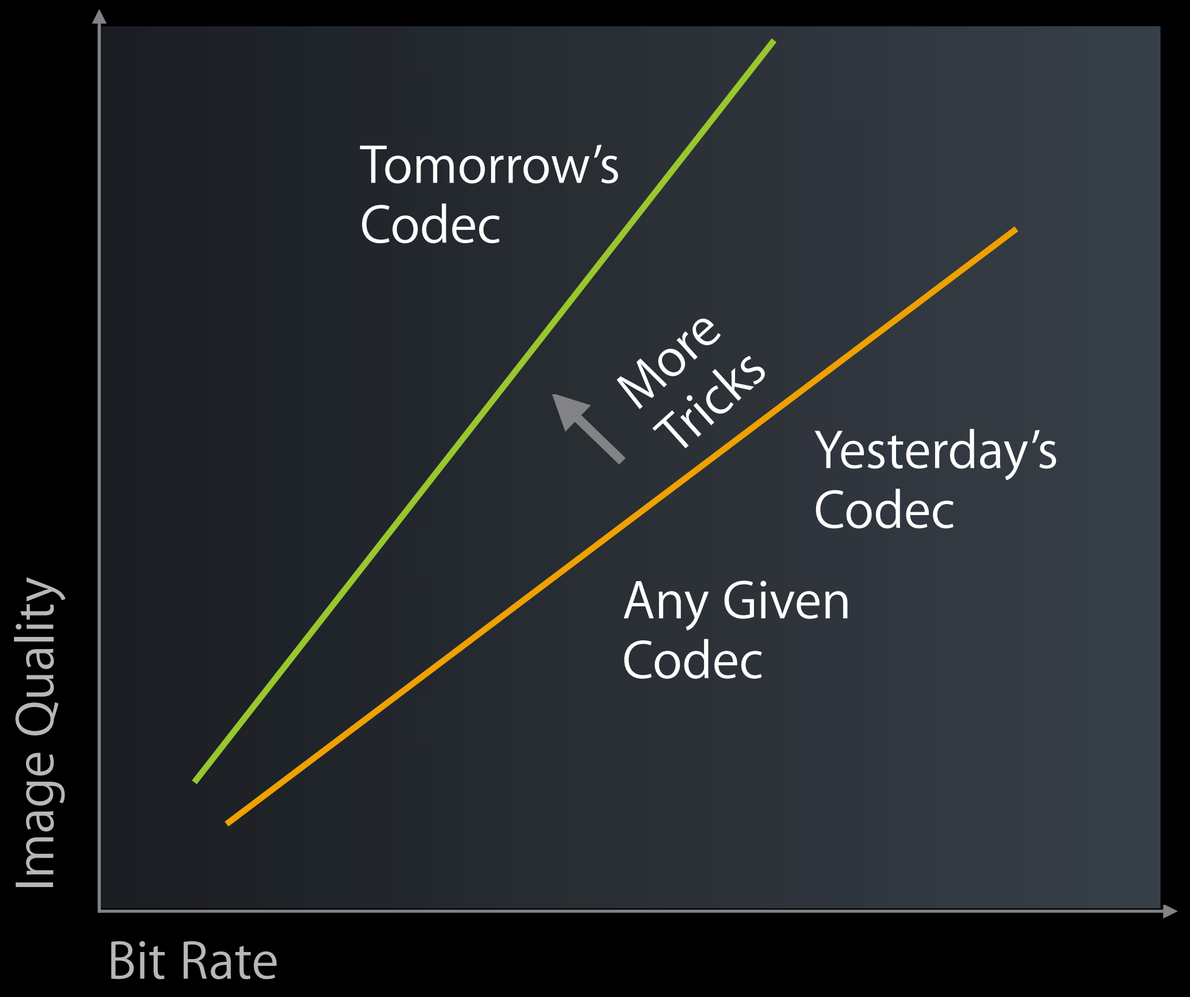
Codecs for the Year of HD

- H.264
  - New premiere delivery codec, scalable from 3G to HD
- HDV
  - MPEG-2 IPB with constant bit-rate, for new FireWire cameras
  - Native support in Final Cut Pro 5
- Apple Intermediate Codec
  - I-frame-only codec for editing HDV in iMovie HD and Final Cut Express HD
- DVCPROHD
  - I-frame-only, DV-style DCT-based codec for HD
- Pixlet
  - I-frame-only
- Uncompressed 4:2:2 8-bit and 10-bit YCbCr
  - I-frame-only

An Interlude on B Frames

http://developer.apple.com/wwdc2005/

What's This all About?



Why Do B Frames Matter?

Suppose you want to compress some video…

Key Frames Only

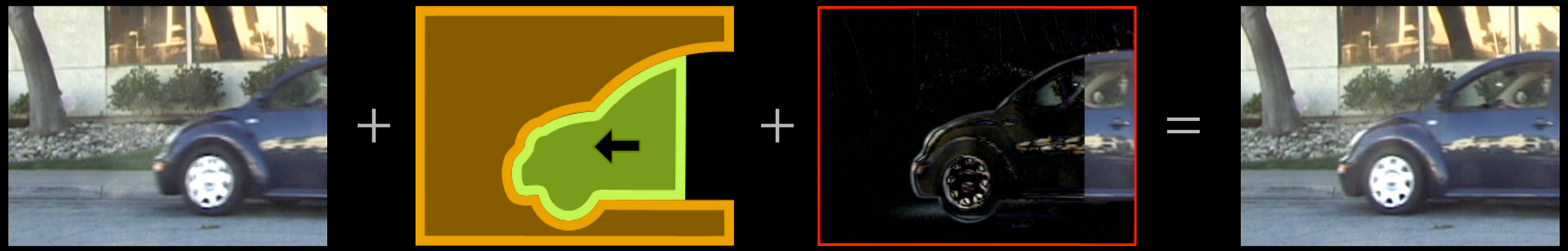
- Every compressed frame is self-contained
- Key frames, sync samples, or I frames (I stands for “Intra”)
- Random access is fast

Time

20K 20K 20K 20K 20K 20K

Similarity

Temporal Prediction



Frame 4 + Motion + Residue = Frame 5

Key Frames and Difference Frames

- Difference frames describe the new image using information from a previous frame
- Also known as P frames (for "predicted")
- Better compression

Time

I 20K

P 5K

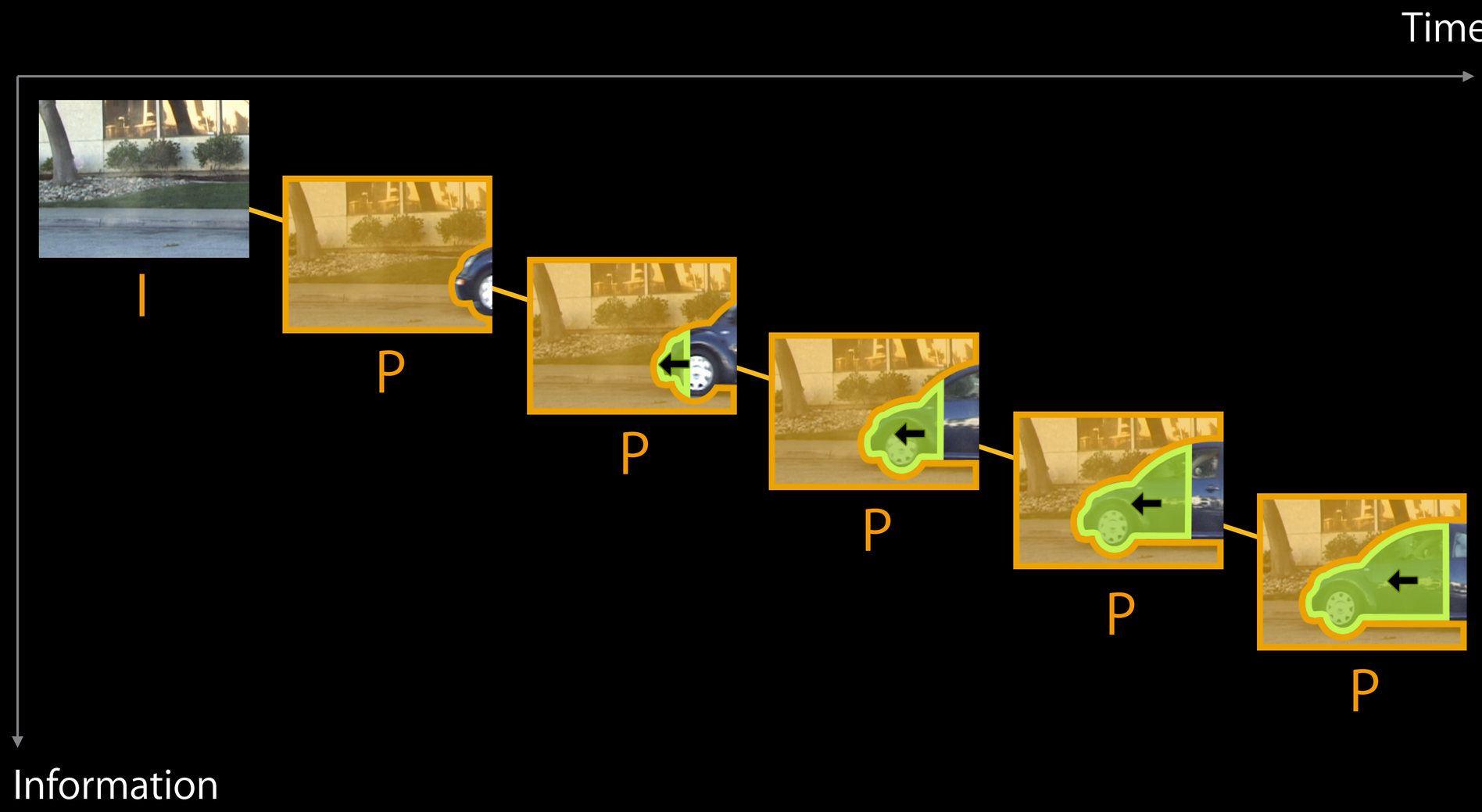
P 5K

P 5K

P 5K

P 5K

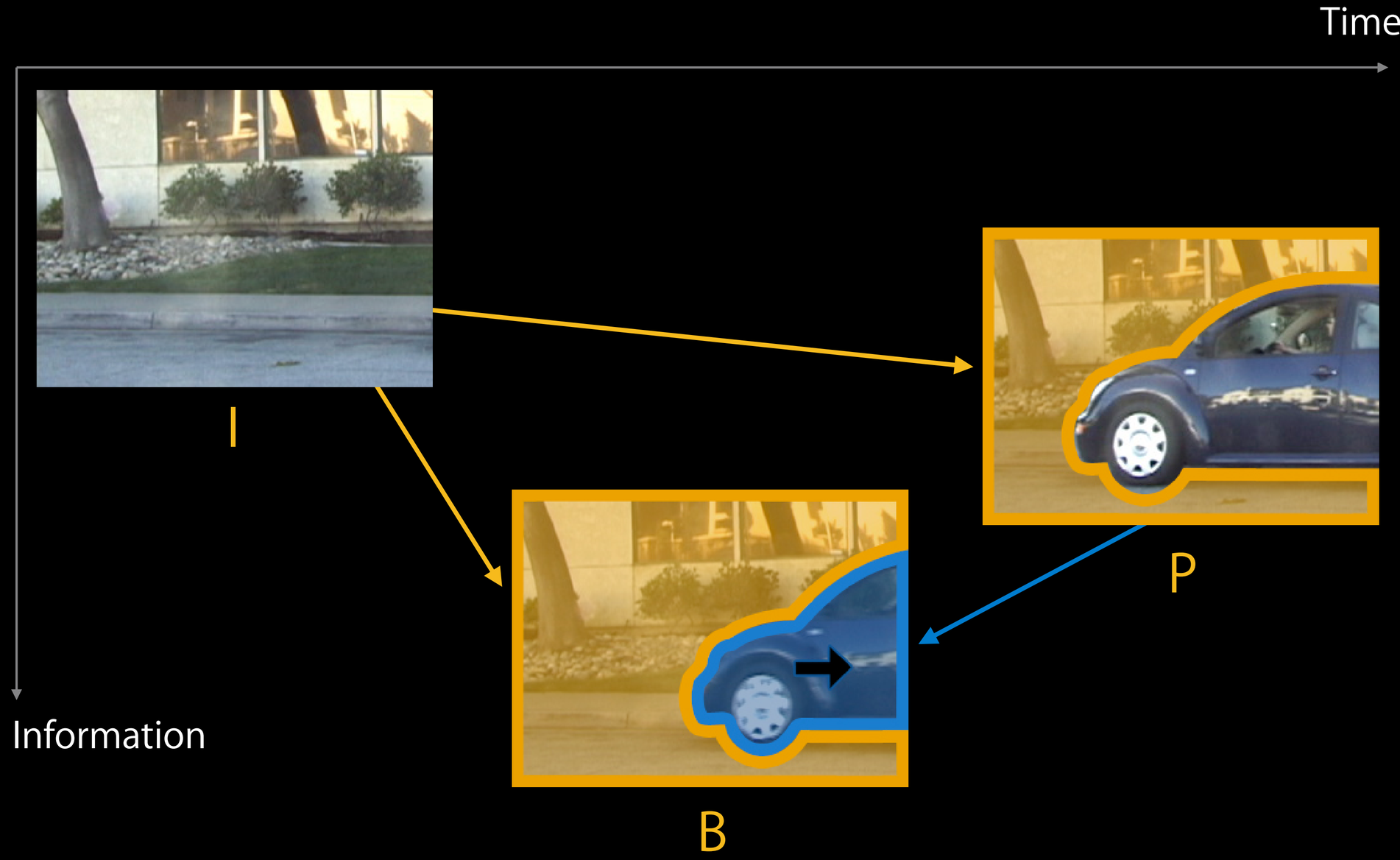
Information Flow



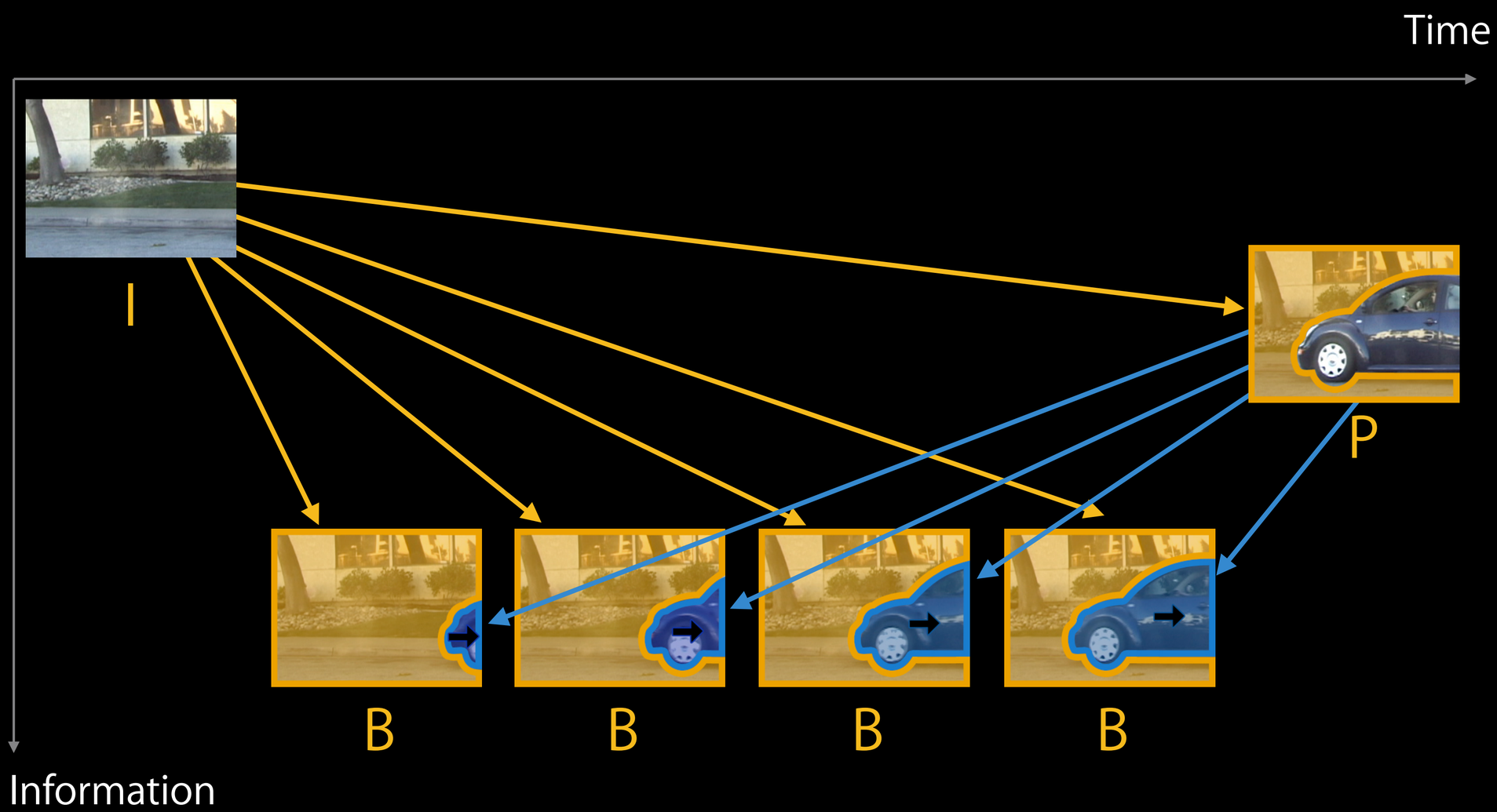
Characteristics of IP

- Better compression than I-only, but
  - Random access can be slow
  - Gradually appearing images constructed incrementally

B Frames: Bidirectional Prediction



B Frames: Bidirectional Prediction



B Frames

- B frames (for ”bidirectional prediction”) refer to information from an I or P frame that will be displayed in the future
- May also be predicted from the previous I or P frame

Time

I
20K

B
1K

B
1K

B
1K

B
1K

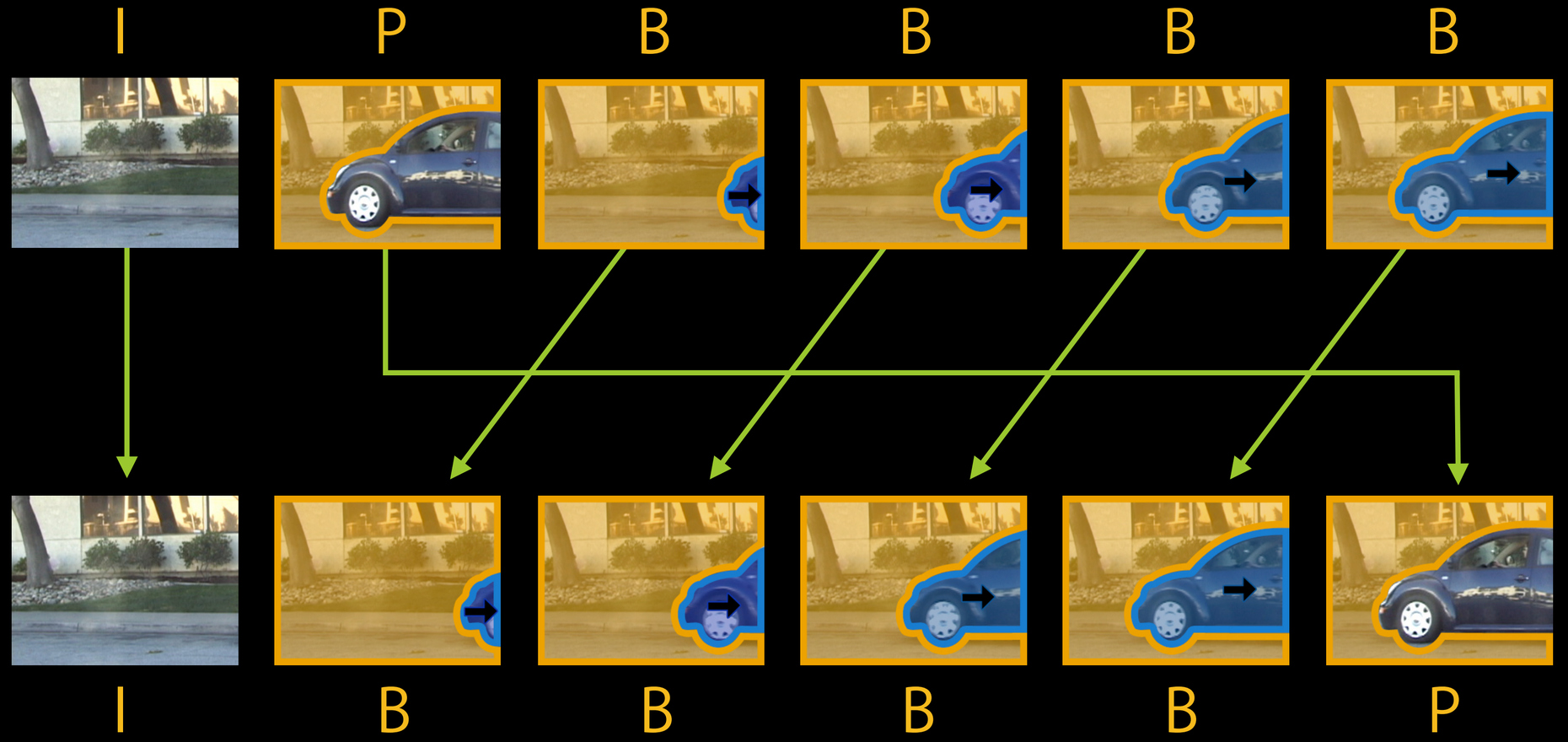
P
10K

Benefits of B Frames

- Better compression when objects appear gradually
- Faster random access: B frames may be skipped (“temporal scalability”)

Decode Order ≠ Display Order

Order in which frames are stored, transmitted, and decoded



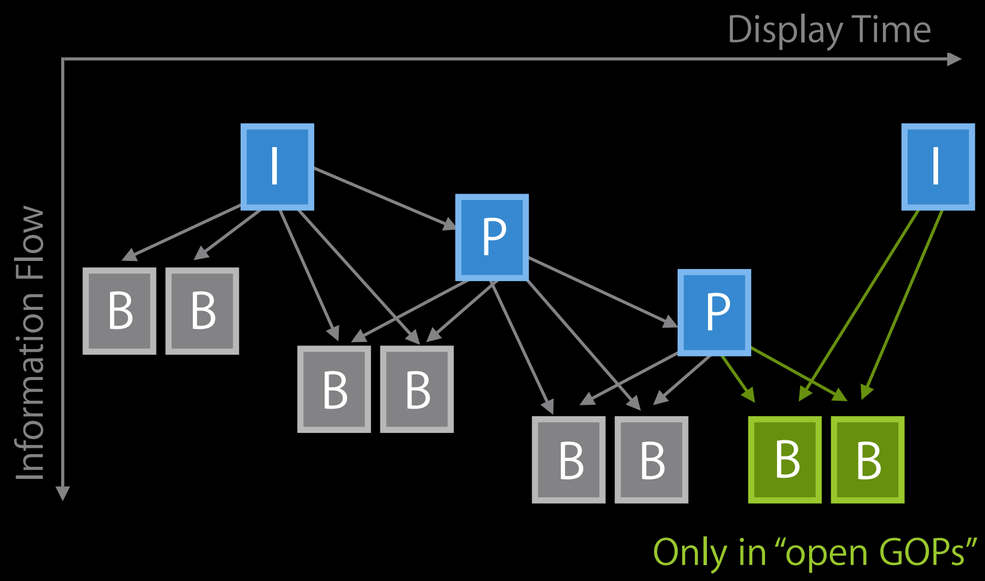
If You Knew All That Already…

- You may also know that you can do all this with a "one-frame delay" if each frame is marked as I, P, or B
- True for MPEG-2
  - Because only one future frame is held at a time
- Not true for H.264
  - Standard allows up to 16 future frames to be held

H.264 Allows Encoder More Flexibility

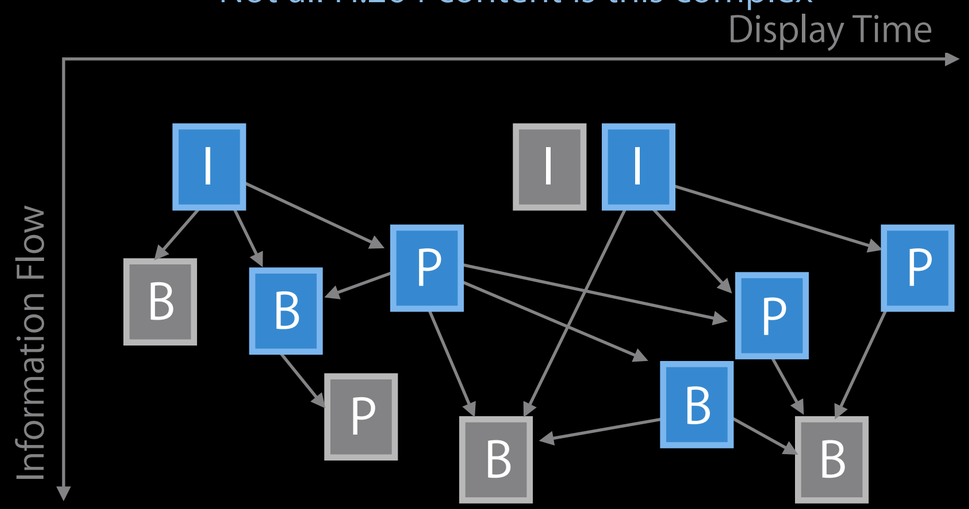
- P frames may use more than one prior frame
- B frames may use more than two prior or future frames
- Not all I frames reset the decoder (all IDR frames do)
- P and B frames may depend on B frames
- Some I and P frames may be dropped

MPEG-1 and -2 Frame Dependencies



H.264 Frame Dependencies

Not all H.264 content is this complex



Rules for H.264

- Don’t derive dependencies from frame types
- Instead, record the following information per frame

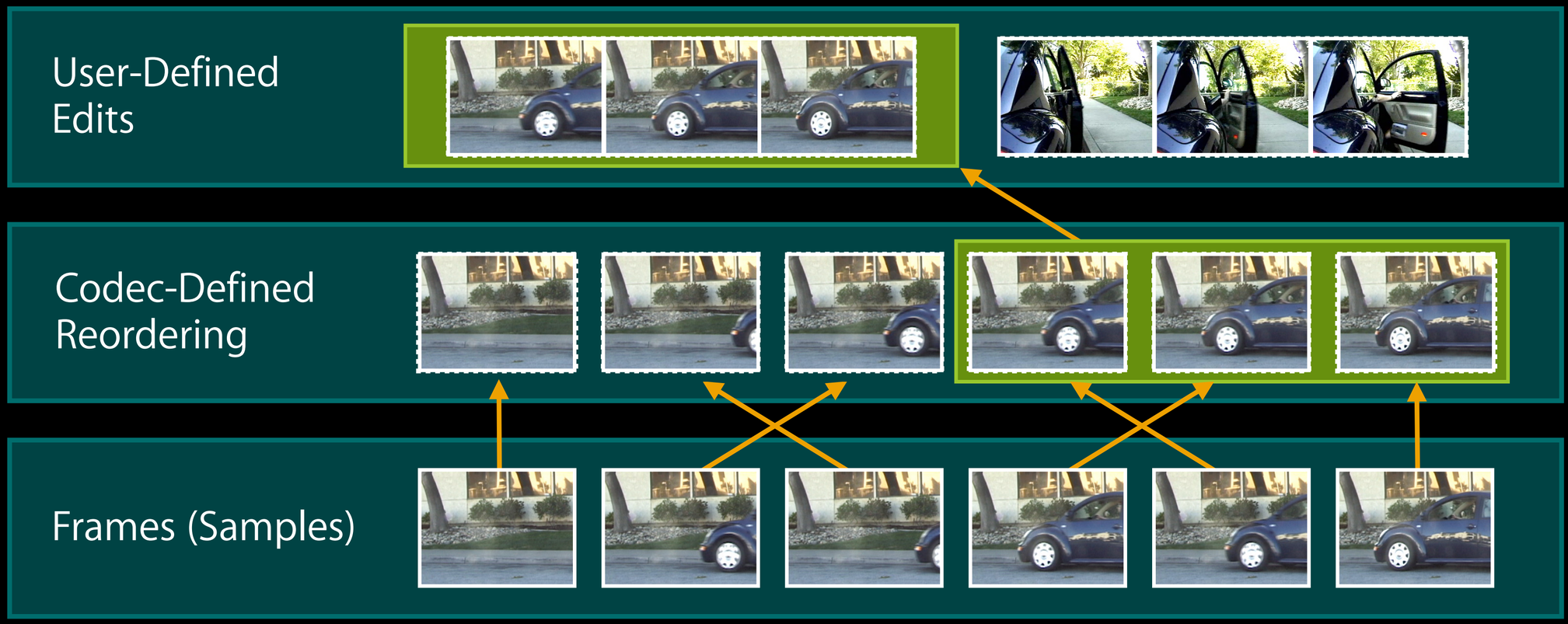
The Moral of This Fable

- Dependencies between frames are getting weirder
  - But it’s all for a good cause
- IPB means frames are stored and decoded in a funny order
- Some convenient rules in MPEG-2 don’t work in H.264

H.264 and the QuickTime API

http://developer.apple.com/wwdc2005/

Anatomy of a Video Track



Each sample has timing and dependency information

API Layers

Other Tracks

User-Defined Edits

Track / Movie

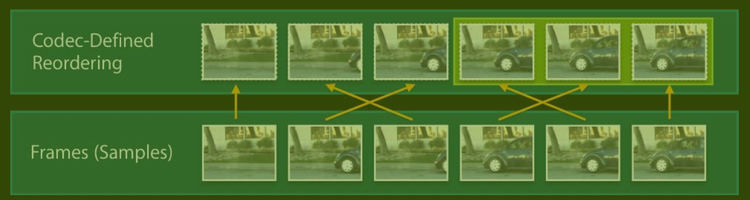
Cut / Copy / Paste (using clipboard)

Track Segment Editing APIs (without clipboard)

Codec-Defined Reordering

Frames (samples)

Media



Sample Manipulation APIs

new

Image Compression and Decompression APIs

new

API Evolution for H.264

- Frame reordering
  - “Display offset” stored for each frame
- Dependency information
  - New sample flags, e.g., “this frame is droppable”

API Evolution Across QuickTime

Away From	Towards	Benefit
Handles	Core foundation-style objects	Retain counting
QuickDraw PixMaps	Core video pixel buffers	Multiple pixel buffers
QuickDraw GWorlds	QuickTime visual contexts	OpenGL hookups
Sound manager	Core audio	High-resolution audio
FSSpecs	Data references	Unicode, long file names
32-bit TimeValues	64-bit TimeValues	High timescales
32-bit sample numbers	64-bit sample numbers	Very long movies

Technote 2140: Modernizing QuickTime Applications
http://developer.apple.com/technotes/tn2005/tn2140.html

Topics We'll Demonstrate

- High-level movie editing
- Multi-pass export from custom sources
- High-level retrieval of decompressed frames from a movie
- Low-level access to compressed frames as stored in a movie
- Low-level image decompression
- Low-level image compression
- Extending QuickTime by writing your own codec

High-Level Movie Editing

David Eldred
High-Definition Engineer

http://developer.apple.com/wwdc2005/

High-Level Movie Editing

- Using clipboard
  - Cut, copy, paste
- Without clipboard
  - “Insert segment” from movie to movie, or track to track
- Saving
  - “Save reference movie”
  - “Save self contained”
- Export
  - to movie, .MP4, .3GPP, etc.
  - Recompress or pass-through

Demo

High-Level Movie Editing

http://developer.apple.com/wwdc2005/

High-Level Movie Editing

- QuickTime handles all that B frame trickiness “under the hood”
- No need to restrict the positions of edits
- Complex compositions work fine
- Can bring together content in many different formats

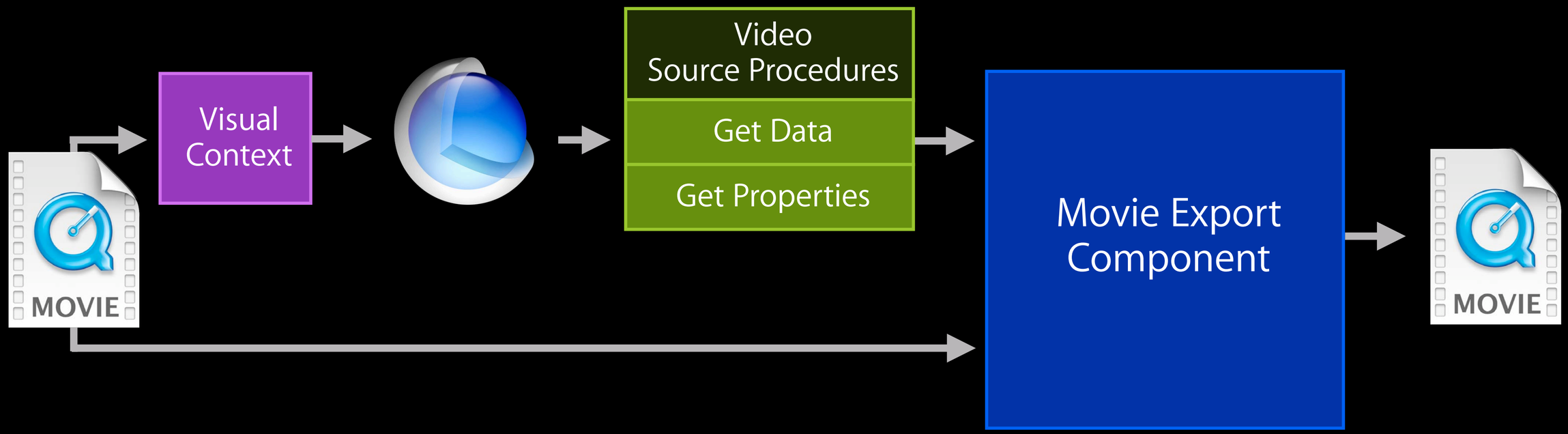
Multi-Pass Export, Custom Source

http://developer.apple.com/wwdc2005/

Multi-Pass Export, Custom Source

- Highest quality compression from application-defined source
- Can let QuickTime handle the compression UI
- You generate the source frames and source audio
- QuickTime handles the rest for you
  - Multi-pass compression (for a multi-pass-aware compressor, e.g. H.264)
  - Movie file writing (or .MP4, .3GP, etc.)
  - Flattening, hinting…

Source Procedures



Demo: CIVideoDemoGL

Multi-pass export from Core Image filter

http://developer.apple.com/wwdc2005/

Multi-Pass Export, Custom Source

- Source procedures are called for every frame of every pass
  - Frames must not change between passes!
- If your rendering is very slow…
  - It may be faster to do a single-pass export to an intermediate movie first

Low-Level Sample Access; Low-Level Image Decompression

Sam Bushell
QuickTime Plumbing

http://developer.apple.com/wwdc2005/

Low-Level Access

- Some applications need to manipulate movies at the frame level
  - More complex editing
  - Effects
  - Fine-grained control over storage or format

Accessing Low-Level Sample Information

- new • To access one frame at a time
  - Use GetMediaSample2
- new • To get information about many frames at once
  - Use CopyMediaMutableSampleTable
- new • Corresponding APIs for adding samples to media
  - AddMediaSample2
  - AddSampleTableToMedia
- Note: older APIs will return errors for movies with B frames

Decompressing Video Frames

- ICMDecompressionSession API
  - You provide a CF Dictionary describing the pixel buffers you want back
  - You push in the compressed frames
  - Your callback is called with CVPixelBuffers and other status messages

Demo: MovieVideoChart

Low-level sample access;

Low-level image decompression

http://developer.apple.com/wwdc2005/

Low-Level APIs

- You must ensure that frames are decoded in the right order
  - Must handle key frames, etc.
  - Must handle reordering
- For high-level access to a movie's decompressed frames, use QTOpenGLTextureContext or QTPixelBufferVisualContext instead

Low-Level Image Compression

http://developer.apple.com/wwdc2005/

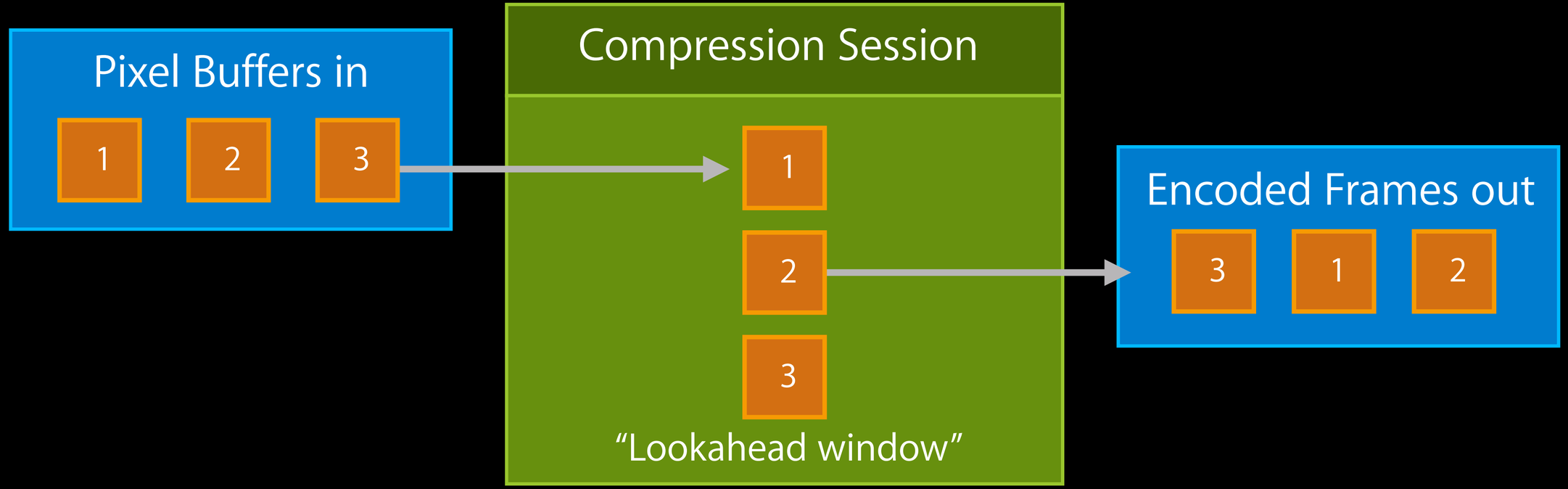
Low-Level Image Compression

- Some applications need to manage video compression
  - e.g. can’t use Movie Export component if target is not a new file

ICM Compression Session

new

- Create with width, height, codec type, and “session options” object
- You push CVPixelBuffers with display times in display order
- Your callback is called with the encoded frames in decode order
- Frames may be reordered during compression



Compressing Live Video to H.264

Sequence Grabber

DV / YUV Frames

Decompression Session

CGBitmapContext

RGB Frames

CGImage

Preview

Demo: CaptureAndCompressIPBMovie

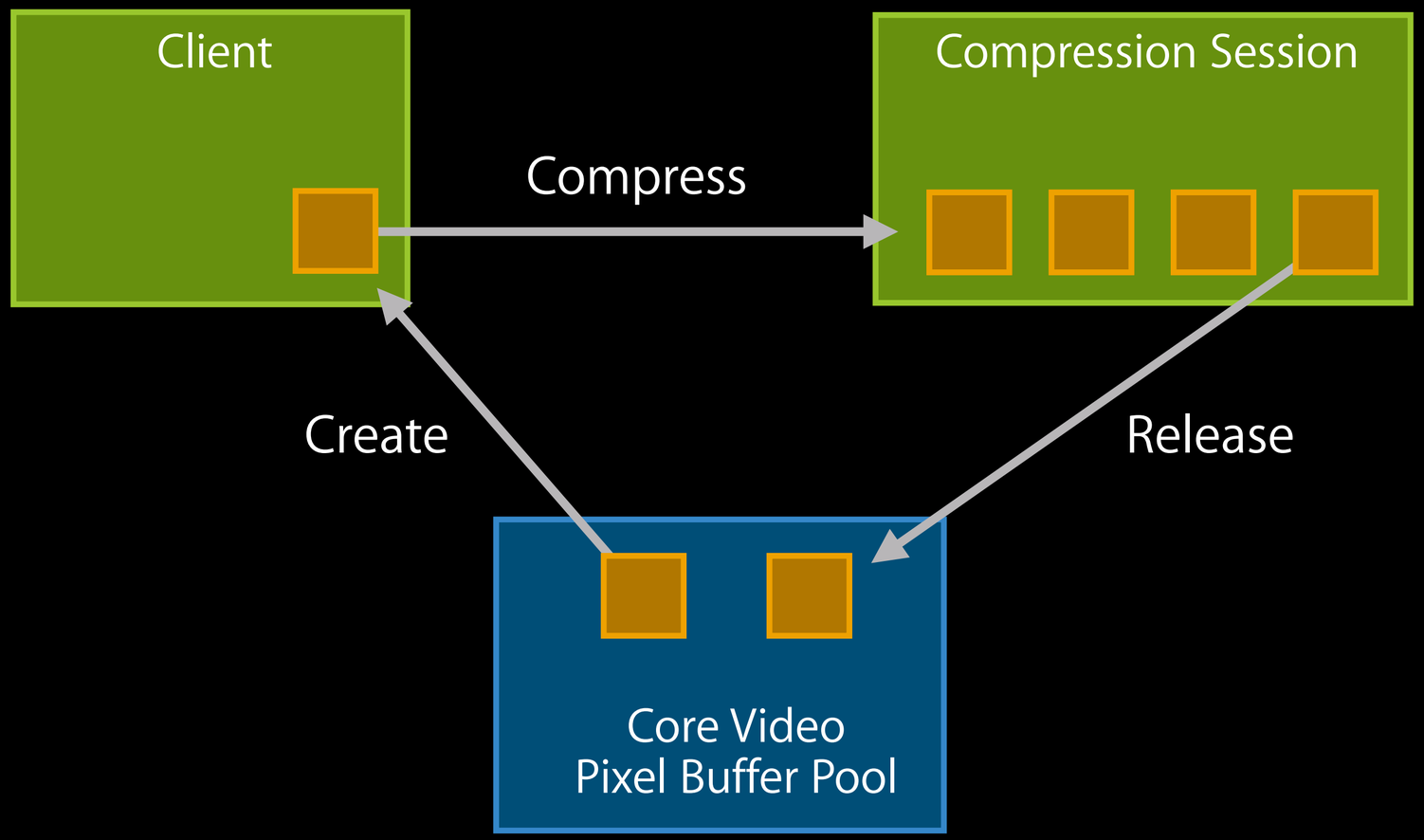
Low-level compression APIs

http://developer.apple.com/wwdc2005/

Low-Level Compression APIs

- H.264 requires new compression APIs
- New compression APIs support all codecs, old and new

Use Pixel Buffer Pools For Efficient Recycling



Writing Codecs

http://developer.apple.com/wwdc2005/

Extending QuickTime by Writing New Codecs

- Need to write two components
  - Image compressor
  - Image decompressor

Demo: ExampleIPBCodec

Writing your own codec

http://developer.apple.com/wwdc2005/

Writing Codecs Is Fun for the Whole Family

- Important functions for compressor
  - ImageCodecPrepareToCompressFrames: describe source pixel buffers
  - ImageCodecEncodeFrame: called with each source frame
  - ImageCodecCompleteFrame: encode this frame, or drop it
- Important functions for decompressor
  - ImageCodecInitialize: report basic characteristics
  - ImageCodecPreflight: negotiate output pixel format, etc
  - ImageCodecBeginBand: classify this frame (droppable? key frame?)
  - ImageCodecDecodeBand: decode a frame to an internal buffer
  - ImageCodecDrawBand: write a frame to an external pixel buffer

Thread-Safety Pledge

- “I promise that every new component I write shall be thread-safe”

Universal Components

- PowerPC applications can only use PowerPC components
- Intel-based applications can only use Intel-based components
- Make your components universal!

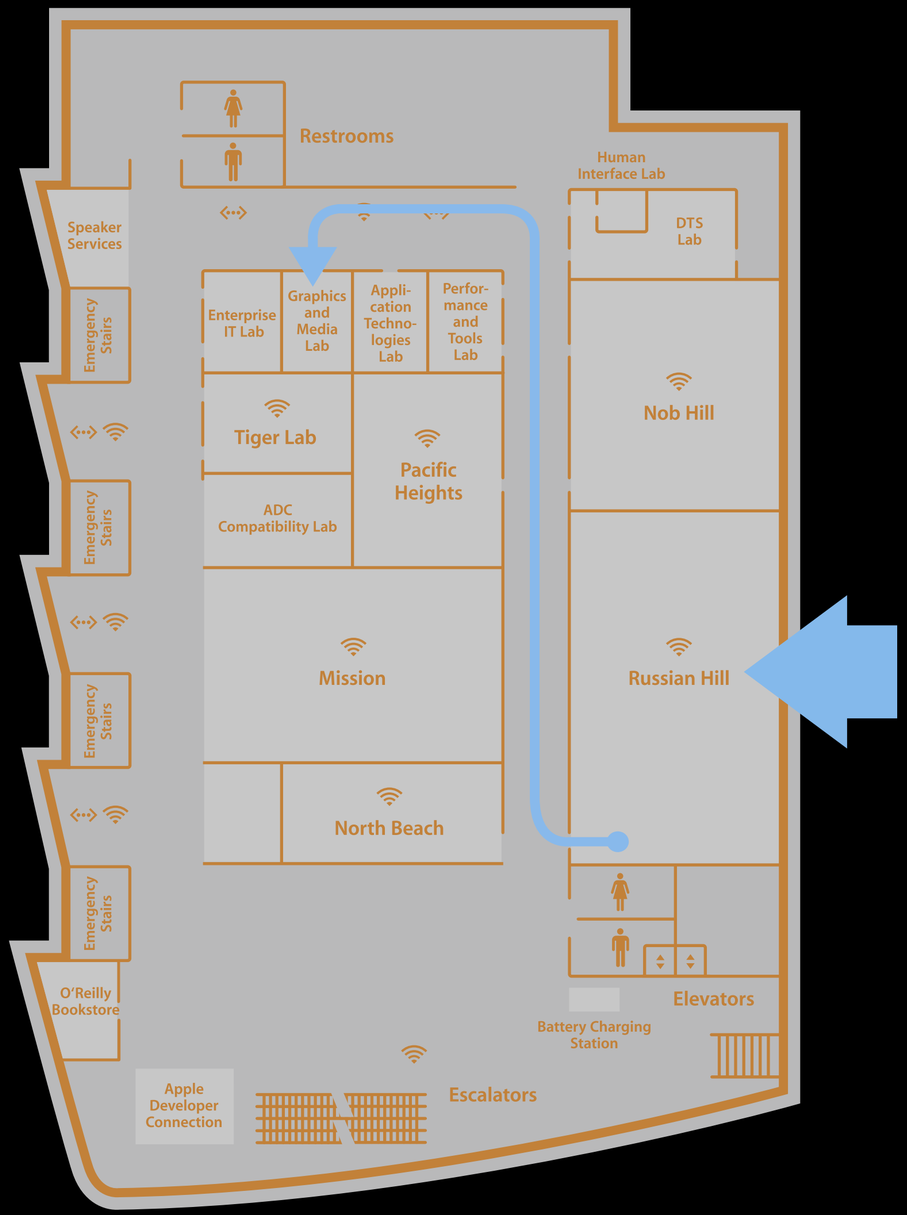
Summary

http://developer.apple.com/wwdc2005/

Samples We've Looked At

- High-level movie editing
  - QTKit
- Multi-pass export from a custom source
  - CIVideoDemoGL
- Low-level sample information and low-level video decompression
  - MovieVideoChart
- Low-level video compression
  - CaptureAndCompressIPBMovie
- Codec
  - ExampleIPBCodec

Visit Us in the Graphics and Media Lab!



Q&A

Tim Cherna
Engineering Manager, QuickTime Video Foundation

WWDC2005

Worldwide Developers Conference